The Dev Lead Trenches

Lessons for Managing Developers

by Chris Tankersley

Edited By Oscar Merida

a php[architect] anthology

Table of Contents

Editor's Note

This book collects almost two-years worth of writings based on Chris Tankersley's experience leading development teams. He first wrote these in his column, also named "The Dev Lead Trenches," for php[architect] magazine. Chris' approach to managing a group of programmers comes from the experiences only another programmer can appreciate. His advice is grounded in an authentic concern for bringing the best out people without treating them as interchangeable cogs. He recognizes the value of well-defined, shared workflows without advocating blind adherence to bureaucratic processes. Whether you're a seasoned lead developer or have just been "promoted" to the role, this collection can help you nurture an expert programming team within your organization.

This book re-organizes his essays thematically, instead of including them in chronological publication order.

- Chapters 1–3 touch upon what the Development Lead role should entail, how to interact with others, and also defines what you should not do.
- Chapters 4–9 look at aspects of managing what your team is tasked with, from project management advice to a workflow for turning feature or bug tickets into deployable code.
- Chapters 10–14 deal with the personnel aspects of finding new hires, assessing individuals, and handling poor performance.
- Chapter 15–18 tackle topics related to your team, or company, culture with advice on what contributes to a positive one and the things to avoid to prevent burnout and toxicity in your workplace.

If you're a newly minted lead, start with chapter one to get your bearings. Otherwise, each chapter can stand alone if you have a specific need for help or insight.

While I've revised the copy from past articles, I did not change the essence of any section. I primarily updated the language to flow as part of a book, replacing mentions of previous articles with links to chapters within this book.

Chapter 1

So Now You're a Team Lead

I never planned on being a team lead.

I'll take that back. If anyone assumes, in their career, they will never lead a team of anything, then they are not really in a career. They are just in a job. Which is fine; you be you.

I never planned on being a team lead when it happened. Way back in 2008 was the first time I was in charge of something, and that something was the web infrastructure and software for an insurance company. I doubt I was an outstanding team lead, though realistically I was just the most senior developer. I began the job as the guy working on the website, but I had a knack for networking as well. After a few years, our applications had grown in complexity and number, and I needed a coworker.

We hired someone who was just getting into programming, and PHP was a relatively new language for him. I was doing more and more architecture work, but I was still in charge of the main web applications and hardware. So I took him under my wing and did my best to show him what I thought were good practices. I wanted to do for him what I had not had done for me. I had gone to school and learned from some excellent teachers but never had someone show me the ropes on a job.

Looking back now, I realize I was mentoring him. Much of it was because I am a very lazy man, so the more I taught him, the fewer questions I had to answer and the more work he could do on his

own. I did honestly want him to succeed, though, because I was overworked and was moving into more of an architectural role. I wanted to leave my code in good hands.

I pushed him to not always automatically agree with me and to go to conferences. I learned how to deal with the politics of the office so we could get suitable hardware. I wanted him to experiment, to learn, and to fail on his own. I don't ever remember him failing but stumbling in a good way. The kind where, when you stand back up, you learned why you stumbled, and not because I was yelling at him for screwing up.

Fast forward to my current job. I was specifically hired to run our web UI team, and part of that is leading an honest-to-goodness team. At this point, I have led a few teams, but my current team is the size I enjoy. We have someone working on the JavaScript frontend, another working on the actual design and layout, myself, and a new hire.

Our new hire is a good friend of mine. I've known her for a few years, and she's far from a junior dev. One day she messaged me that she was having some problems with a pull request. Something was just not working right when she was pushing code up. I will admit we have kind of a janky workflow at the moment, but looking in GitHub, she had somehow duplicated all of the commits in master and pushed it up to the central repository. Pull requests that were waiting now looked all sorts of funky.

I logged in, pulled down another copy of the master from our other coworker, and did a force push back up. All fixed. She and I then set to work figuring out what happened.

I wasn't mad. Thankfully, the distributed nature of Git makes it incredibly easy to have multiple backups of repositories sitting around, so I was never in a state of panic. People joke all the time about deleting a production database and the fallout, but this did not even come close. I *have* deleted production databases before—during the workday. S**t hits the fan *really* quick. We took it in stride, and we figured out what happened.

The Accidental Team Lead

At some point in your programming career, you will become a team lead, either formally or informally. Your title might be *Lead Developer*, *Dev Lead*, *Programming Manager*, or even *Programmer*. If you find yourself suddenly charged with making decisions for a group of people, congratulations! You are now a Team Lead.

I want to do for you what I did for the first coworker I led all those years ago. No one taught me how to be a team lead, just like no one taught me how to program in PHP. As with any job, there will be differences between what each team lead does, but I feel all team leads have a core set of responsibilities they will need to do. Most of them will have nothing at all to do with programming and everything to do with people. I am still learning myself, and I stumble every so often. As long as you are willing to learn and learn from your mistakes, you will be a great team lead.

Throughout the subsequent articles in this column, I will touch on the following ideas and more. Since this is my first time writing about being a team lead, and hopefully, you, as a reader, want to learn right away, here are a few of the things you need to consider as a team lead.

Communication Is Key

I am a remote developer, so this hits a few deep points with me, but as a team lead you need to make sure communication is open between you and the people you work with. You will be the one in charge of making sure deadlines are understood both between the team and upper management. You will be the one that needs to know when problems arise. You need to know when your team is blocked and needs something.

Don't be the kind of lead that only talks to the team at meetings. A good lead will be right there with the rest of the team, talking to them on a daily basis. Daily standup meetings are helpful, but you want to be available and listen when there are problems or there are needs. Make sure your team has the tools to communicate. Have e-mail, chat, and verbal conversations when appropriate.

Many programmers are introverts and find it hard to talk to groups of people. One-on-one communication may be more comfortable. As a team lead you will need to overcome that in whatever way works for you. You need to listen. Make sure your team knows they can talk to you—and even question you.

Communication goes both ways. You need to work with your team to translate management's needs and wishes to the team. Effective written and verbal communication is critical.

Above all, be respectful in your communication. We are all adults and deserve to be treated as such, even if the other person is wrong.

Help The Team Grow

It will be your job as a team lead to make sure your team is growing. Well-oiled development teams are the ones with the ability to use whatever best practices are in the marketplace at the time. Your team will be stronger knowing they can grow in their skills and not stagnate. Experimentation is vital, even if you do not dedicate specific time to research and development.

You need to help the team grow. Work with upper management to make sure a training budget is available. Find a way to send team members to conferences, either as attendees or speakers depending on experience. There are plenty of regional conferences around the US, and Europe has a handful of excellent conferences to attend.

Encourage reading and watching videos on new topics. Sign up for services like Laracasts or whatever would be equivalent for your programming stack. Find new tools, and let your team find new tools that make their jobs easier and more productive. Allow the team to find what works for them.

While you don't have to accept every single idea under the sun, at least let the team try new things, and have chances to learn new things.

Shield the Team

You will be the first line of defense from management. That sounds harsh, but the reality is that even the best managers are still managers.

I work with a great team and have a fantastic manager, but there are times when feature requests and bugs make it onto our issue board which, well, shouldn't be there. In meetings, you are a representative of your team and will be the one dealing with requests.

Learning to work with management and keep them at arm's length from the rest of the team is a good thing. You are a liaison, but also a filter. You get to deal with management, schedules, and budgets, so other people on the team do not have to. Part of your job is now to keep your team focused and productive. I sit through an entire Monday's worth of meetings, which I would not wish on anyone else on my team.

If features are not going to be met by the deadline, you are the one informing the other teams and management. You might be on the team, but as its lead, you are the voice of the team.

Stand Up for the Team

As a liaison between upper management and the team, it is also your responsibility to stand up for your team. If deadlines are not reasonable, learning to push back is a must. Put on your project management hat and work with management and teams to find solutions, but always keep in mind your team is made up of people who assume you are working with, and for, them.

You are the one that will need to make requests for hardware, for conferences, and for better deadlines. Make sure praise is not kept inside the team but shared upward.

A good team lead will work to remove roadblocks for the team whenever possible. It might be hard, and you will not always win, but you must try.

Let's Be on a Team

I want to help you grow as a team lead, whether you took a job as a team lead want to do better, or were like me and all of a sudden in charge of other people. Ultimately, it is not as scary as one might think, but being a team lead may mean getting outside of your comfort zone.

As these columns continue, I will expand upon the few ideas I mentioned before and more. I want to share my stories with you as an actual team lead on various teams so you can learn from my mistakes. I want to give you the tools to effectively manage your team and to help your team members grow.

As a sneak peek for the next chapter, consider this—don't tell your coworker a monkey could do a better job than them.

Index

D

E

G

H

K

L

M

O

P

Q

R

S

T

U

W

php[architect] Books

The php[architect] series of books cover topics relevant to modern PHP programming. We offer our books in both print and digital formats. Print copy price includes free shipping to the US. Books sold digitally are available to you DRM-free in PDF, ePub, or Mobi formats for viewing on any device that supports these.

To view the complete selection of books and order a copy of your own, please visit: *http://phparch.com/books/*.

- **Web Scraping with PHP, 2nd Edition**
  By Matthew Turland
  ISBN: 978-1940111674

- **Security Principles for PHP Applications**
  By Eric Mann
  ISBN: 978-1940111612

- **Docker for Developers, 2nd Edition**
  By Chris Tankersley
  ISBN: 978-1940111568 (Print edition)

- **What's Next? Professional Development Advice**
  Edited by Oscar Merida
  ISBN: 978-1940111513

- **Functional Programing in PHP, 2nd Edition**
  By: Simon Holywell
  ISBN: 978-1940111469

- **Web Security 2016**
  Edited by Oscar Merida
  ISBN: 978-1940111414

- **Building Exceptional Sites with WordPress & Thesis**
  By Peter MacIntyre
  ISBN: 978-1940111315

- **Integrating Web Services with OAuth and PHP**
  By Matthew Frost
  ISBN: 978-1940111261

- **Zend Framework 1 to 2 Migration Guide**
  By Bart McLeod
  ISBN: 978-1940111216

- **XML Parsing with PHP**
  By John M. Stokes
  ISBN: 978-1940111162

- **Zend PHP 5 Certification Study Guide, Third Edition**
  By Davey Shafik with Ben Ramsey
  ISBN: 978-1940111100

- **Mastering the SPL Library**
  By Joshua Thijssen
  ISBN: 978-1940111001